```
                        title   'CP/M V3.0 Loader'


                ;       Copyright (C) 1982
                ;       Digital Research
                ;       Box 579, Pacific Grove
                ;       California, 93950

                ;  Revised:
                ;    01 Nov 82  by Bruce Skidmore

0000 =          base    equ     $
FF00 =          abase   equ     base-0100h

000D =          cr      equ     0dh
000A =          lf      equ     0ah

FF5C =          fcb     equ     abase+005ch     ;default FCB address
FF80 =          buff    equ     abase+0080h     ;default buffer address

                ;
                ;       System Equates
                ;
000D =          resetsys        equ     13      ;reset disk system
0009 =          printbuf        equ     09      ;print string
000F =          open$func       equ     15      ;open function
0014 =          read$func       equ     20      ;read sequential
001A =          setdma$func     equ     26      ;set dma address
                ;
                ;       Loader Equates
                ;
FF80 =          comtop  equ     abase+80h
FF81 =          comlen  equ     abase+81h
FF82 =          bnktop  equ     abase+82h
FF83 =          bnklen  equ     abase+83h
FF84 =          osentry         equ     abase+84h

                        cseg

0000 318101             lxi     sp,stackbot

0003 CD000A             call    bootf           ;first call is to Cold Boot

0006 0E0D               mvi     c,resetsys      ;Initialize the System
0008 CD8D01             call    bdos

000B 0E09               mvi     c,printbuf      ;print the sign on message
000D 112501             lxi     d,signon
0010 CD8D01             call    bdos

0013 0E0F               mvi     c,open$func     ;open the CPM3.SYS file
0015 11AB00             lxi     d,cpmfcb
0018 CD8D01             call    bdos
001B FEFF               cpi     0ffh
001D 11CF00             lxi     d,openerr
0020 CAA200             jz      error
```

CP/M-PLUS V3.0
COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. # CP3-135

```
                        
0023 1180FF             lxi     d,buff
0026 CD8F00             call    setdma$proc
                        
0029 CD9500             call    read$proc       ;read the load record
                        
002C 2180FF             lxi     h,buff
002F 118101             lxi     d,mem$top
0032 0E06               mvi     c,6
                cloop:
0034 7E                 mov     a,m
0035 12                 stax    d
0036 13                 inx     d
0037 23                 inx     h
0038 0D                 dcr     c
0039 C23400             jnz     cloop
                        
003C CD9500             call    read$proc       ;read display info
                        
003F 0E09               mvi     c,printbuf      ;print the info
0041 1180FF             lxi     d,buff
0044 CD8D01             call    bdos
                        
                ;
                ;       Main System Load
                ;
                        
                ;
                ;       Load Common Portion of System
                ;
0047 3A8201             lda     res$len
004A 67                 mov     h,a
004B 3A8101             lda     mem$top
004E CD7300             call    load
                ;
                ;       Load Banked Portion of System
                ;
0051 3A8401             lda     bank$len
0054 B7                 ora     a
0055 CA5F00             jz      execute
0058 67                 mov     h,a
0059 3A8301             lda     bank$top
005C CD7300             call    load
                ;
                ;       Execute System
                ;
                execute:
005F 215DFF             lxi     h,fcb+1
0062 7E                 mov     a,m
0063 FE24               cpi     '$'
0065 C26F00             jnz     execute$sys
0068 23                 inx     h
0069 7E                 mov     a,m
006A FE42               cpi     'B'
006C CCA900             cz      break
                execute$sys:
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
  113  0071 318501          lxi     sp,osentry$adr
  114  0072 C9              ret
  115
  116                  ;
  117                  ;       Load Routine
  118                  ;
  119                  ;       Input:   A = Page Address of load top
  120                  ;                H = Length in pages of module to read
  121                  ;
  122                  load:
  123  0073 B7              ora     a       ;clear carry
  124  0074 57              mov     d,a
  125  0075 1E00            mvi     e,0
  126  0077 7C              mov     a,h
  127  0078 17              ral
  128  0079 67              mov     h,a     ;h = length in records of module
  129                  loop:
  130  007A EB              xchg
  131  007B 0180FF          lxi     b,-128
  132  007E 09              dad     b       ;decrement dma address by 128
  133  007F EB              xchg
  134  0080 D5              push    d
  135  0081 E5              push    h
  136  0082 CD8F00          call    setdma$proc
  137  0085 CD9500          call    read$proc
  138  0088 E1              pop     h
  139  0089 D1              pop     d
  140  008A 25              dcr     h
  141  008B C27A00          jnz     loop
  142  008E C9              ret
  143
  144                  ;
  145                  ;       Set DMA Routine
  146                  ;
  147                  setdma$proc:
  148  008F 0E1A            mvi     c,setdma$func
  149  0091 CD8D01          call    bdos
  150  0094 C9              ret
  151
  152                  ;
  153                  ;       Read Routine
  154                  ;
  155                  read$proc:
  156  0095 0E14            mvi     c,read$func     ;Read the load record
  157  0097 11AB00          lxi     d,cpmfcb        ;into address 80h
  158  009A CD8D01          call    bdos
  159  009D B7              ora     a
  160  009E 11FA00          lxi     d,readerr
  161  00A1 C8              rz
  162                  ;
  163                  ;       Error Routine
  164                  ;
  165                  error:
  166  00A2 0E09            mvi     c,printbuf      ;print error message
  167  00A4 CD8D01          call    bdos
  168  00A7 F3              di
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
00AB 76              hlt

                     break:
00A9 FF              db      0ffh
00AA C9              ret

                     cpmfcb:
00AB 0043504D33      db      0,'CPM3    SYS',0,0,0,0,0,0
00BD 0000000000      dw      0,0,0,0,0,0,0,0

                     openerr:
00CF 0D0A            db      cr,lf
00D1 43504D4C44      db      'CPMLDR error:  failed to open CPM3.SYS'
00F7 0D0A24          db      cr,lf,'$'

                     readerr:
00FA 0D0A            db      cr,lf
00FC 43504D4C44      db      'CPMLDR error:  failed to read CPM3.SYS'
0122 0D0A24          db      cr,lf,'$'

                     signon:
0125 0D              db      cr
0126 0A0A0A0A0A      db      lf,lf,lf,lf,lf,lf,lf,lf,lf,lf,lf,lf,lf
0132 0A0A0A0A0A      db      lf,lf,lf,lf,lf,lf,lf,lf,lf,lf,lf,lf,lf
013E 43502F4D20      db      'CP/M V3.0 Loader',cr,lf
0150 436F707972      db      'Copyright (C) 1982, Digital Research'
0174 0D0A24          db      cr,lf,'$'

0177 3032313138      db      '021182',0,0,0,0
                     stackbot:

                     mem$top:
0181                 ds      1
                     res$len:
0182                 ds      1
                     bank$top:
0183                 ds      1
                     bank$len:
0184                 ds      1
                     osentry$adr:
0185                 ds      2

                     ;       title   'CP/M 3.0 LDRBDOS Interface, Version 3.1 Nov, 1982'
                     ;*****************************************************************
                     ;*****************************************************************
                     ;**                                                             **
                     ;**  B a s i c   D i s K   O p e r a t i n g   S y s t e m      **
                     ;**                                                             **
                     ;**           I n t e r f a c e   M o d u l e                   **
                     ;**                                                             **
                     ;*****************************************************************
                     ;*****************************************************************
                     ;
                     ;       Copyright (c) 1978, 1979, 1980, 1981, 1982
                     ;       Digital Research
                     ;       Box 579, Pacific Grove
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                ;       California
                ;
                ;        Nov 1982
                ;
                ;
                ;       equates for non graphic characters
                ;
007F =          rubout equ     7fh     ; char delete
0009 =          tab    equ     09h     ; tab char
000D =          cr     equ     0dh     ; carriage return
000A =          lf     equ     0ah     ; line feed
0008 =          ctlh   equ     08h     ; backspace


                ;
0187 0000000000serial: db      0,0,0,0,0,0
                ;
                ;       Enter here from the user's program with function number in c,
                ;       and information address in d,e
                ;

                bdos:
                bdose:                                  ; Arrive here from user programs
018D EB22DF08EB         xchg! shld info! xchg           ; info=de, de=info

0192 79FE0EDAA5         mov a,c! cpi 14! jc bdose2
0198 32E208             sta fx                          ; Save disk function #
019B AF32B708           xra a! sta dircnt
019F 3ADE0832BE         lda seldsk! sta olddsk          ; Save seldsk

                bdose2:
01A5 7B32BA08           mov a,e! sta linfo              ; linfo = low(info) - don't equ
01A9 2100002209         lxi h,0! shld aret              ; Return value defaults to 0000
01AF 22E108             shld resel                      ; resel = 0
                                                ; Save user's stack pointer, set to local stack
01B2 39224002           dad sp! shld entsp              ; entsp = stackptr

01B6 319202             lxi sp,lstack                   ; local stack setup

01B9 214E08             lxi h,goback                    ; Return here after all functions
01BC E5                 push h                          ; jmp goback equivalent to ret
01BD 79FE32D2CA         mov a,c! cpi nfuncs! jnc high$fxs ; Skip if invalid #
01C3 4B                 mov c,e                         ; possible output character to c
01C4 21DC01C3CF         lxi h,functab! jmp bdos$jmp

                        ; look for functions 100 ->
                high$fxs:
01CA DE64DA9B06         sbi 100! jc lret$eq$ff          ; Skip if function < 100

                bdos$jmp:

01CF 5F1600             mov e,a! mvi d,0                ; de=func, hl=.ciotab
01D2 19195E2356         dad d! dad d! mov e,m! inx h! mov d,m ; de=functab(func)
01D7 2ADF08             lhld info                       ; info in de for later xchg
01DA EBE9               xchg! pchl                      ; dispatched
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                ;       dispatch table for functions

                functab:
01DC E802E802BB         dw      func$ret, func1, func2, func3
01E4 E802E802E8         dw      func$ret, func$ret, func6, func$ret
01EC E802DF02E8         dw      func$ret, func9, func10, func11
000C =          diskf   equ     ($-functab)/2   ; disk funcs
01F4 E802000817         dw      func12,func13,func14,func15
01FC E802E802E8         dw      func16,func17,func18,func19
0204 3B08E802E8         dw      func20,func21,func22,func23
020C E802410847         dw      func24,func25,func26,func27
0214 E802E802E8         dw      func28,func29,func30,func31
021C E802E802E8         dw      func32,func33,func34,func35
0224 E802E802E8         dw      func36,func37,func38,func39
022C E802E802E8         dw      func40,func42,func43
0232 E802E802E8         dw      func44,func45,func46,func47
023A E802E802E8         dw      func48,func49,func50
0032 =          nfuncs  equ     ($-functab)/2

0240            entsp:  ds      2       ; entry stack pointer

                        ;       40 level stack

0242 C7C7C7C7C7         dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
0252 C7C7C7C7C7         dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
0262 C7C7C7C7C7         dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
0272 C7C7C7C7C7         dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
0282 C7C7C7C7C7         dw      0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h,0c7c7h
                lstack:


                page
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                        title  'CP/M 3.0 LDRBDOS Interface, Version 3.1 July, 1982'
                    ;*****************************************************************
                    ;*****************************************************************
                    ;**                                                             **
                    ;**  B a s i c   D i s K   O p e r a t i n g   S y s t e m      **
                    ;**                                                             **
                    ;**              C o n s o l e   P o r t i o n                  **
                    ;**                                                             **
                    ;*****************************************************************
                    ;*****************************************************************
                    ;
                    ;       July, 1982
                    ;
                    ;
                    ;       console handlers
                    
                    conout:
                                            ;compute character position/write console char from C
                                            ;compcol = true if computing column position
0292 3AEE02B7C2             lda compcol! ora a! jnz compout
                                            ;write the character, then compute the column
                                            ;write console character from C
0299 C5                     push b                      ;recall/save character
029A CD0C0A                 call conoutf                ;externally, to console
029D C1                     pop b                       ;recall the character
                      compout:
029E 79                           mov a,c               ;recall the character
                                                        ;and compute column position
029F 21DA08                       lxi h,column          ;A = char, HL = .column
02A2 FE7FC8                       cpi rubout! rz        ;no column change if nulls
02A5 34                           inr m                 ;column = column + 1
02A6 FE20D0                       cpi ' '! rnc          ;return if graphic
                                                        ;not graphic, reset column position
02A9 35                           dcr m                 ;column = column - 1
02AA 7EB7C8                       mov a,m! ora a! rz    ;return if at zero
                                                        ;not at zero, may be backspace or eol
02AD 79                           mov a,c               ;character back to A
02AE FE08C2B502                   cpi ctlh! jnz notbacksp
                                                        ;backspace character
02B3 35                           dcr m                 ;column = column - 1
02B4 C9                           ret
                    
                      notbacksp:
                                                        ;not a backspace character, exit
02B5 FE0AC0                       cpi lf! rnz           ;return if not
                                                        ;end of line, column = 0
02B8 3600                         mvi m,0               ;column = 0
02BA C9                           ret
                    ;
                    ;
                    tabout:
                                                        ;expand tabs to console
02BB 79FE09C292             mov a,c! cpi tab! jnz conout    ;direct to conout if not
                                                        ;tab encountered, move to next tab pos
                      tab0:
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
 372  02C1 0E20CD9202      mvi c,' '! call conout          ;another blank
 373  02C6 3ADA08E607      lda column! ani 111b            ;column mod 8 = 0 ?
 374  02CB C2C102          jnz tab0                        ;back for another if not
 375  02CE C9              ret
 376                  ;
 377                  print:
 378                                                       ;print message until M(BC) = '$'
 379  02CF 21DB08          LXI H,OUTDELIM
 380  02D2 0ABEC8          ldax b! CMP M! rz               ;stop on $
 381                                                       ;more to print
 382  02D5 03C54F          inx b! push b! mov c,a          ;char to C
 383  02D8 CDBB02          call tabout                     ;another character printed
 384  02DB C1C3CF02        pop b! jmp print
 385                  ;
 386                  ;
 387  02BB =          func2: equ     tabout
 388                                      ;write console character with tab expansion
 389                  ;
 390                  func9:
 391                                                       ;write line until $ encountered
 392  02DF EB              xchg                            ;was lhld info
 393  02E0 4D44            mov c,l! mov b,h                ;BC=string address
 394  02E2 C3CF02          jmp print                       ;out to console
 395                  ;
 396                  sta$ret:
 397                                                       ;store the A register to aret
 398  02E5 320903          sta aret
 399                  func$ret:
 400  02E8 C9              ret                    ;jmp goback (pop stack for non cp/m functions)
 401                  ;
 402                  setlret1:
 403                                                       ;set lret = 1
 404  02E9 3E01C3E502      mvi a,1! jmp sta$ret
 405                  ;
 406  02EB =          func1: equ     func$ret
 407                  ;
 408  02EB =          func3: equ     func$ret
 409                  ;
 410  02EB =          func6: equ     func$ret
 411                  ;
 412  02EB =          func10:        equ     func$ret
 413  02EB =          func11:        equ     func$ret
 414                  ;
 415                  ;       data areas
 416                  ;
 417
 418
 419  02EE 00         compcol:db     0       ;true if computing column position
 420                  ;       end of BDOS Console module
 421
 422                  ;************************************************************************
 423                  ;************************************************************************
 424                  ;
 425                  ;       Error Messages
 426
 427  0024 =          md     equ     24h
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                        
02EF 0D0A42444Ferr$msg: db      cr,lf,'BDOS ERR: ',md
02FC 53656C6563err$select: db   'Select',md
0303 5065726D2Eerr$phys: db     'Perm.',md

                ;*****************************************************************
                ;*****************************************************************
                ;
                ;       common values shared between bdosi and bdos

0309            aret:   ds      2       ; address value to return
0309 =          lret    equ     aret    ; low(aret)

                ;*****************************************************************
                ;*****************************************************************
                ;**                                                             **
                ;**  b a s i c   d i s k   o p e r a t i n g   s y s t e m      **
                ;**                                                             **
                ;*****************************************************************
                ;*****************************************************************

                ;       literal constants

00FF =          true    equ     0ffh    ; constant true
0000 =          false   equ     000h    ; constant false
FFFF =          enddir  equ     0ffffh  ; end of directory
0001 =          byte    equ     1       ; number of bytes for 'byte' type
0002 =          word    equ     2       ; number of bytes for 'word' type

                ;       fixed addresses in low memory

0080 =          tbuff   equ     0080h   ; default buffer location

                ;       error message handlers

                sel$error:
                                        ; report select error
030B 01EF02             lxi b,err$msg
030E CDCF02             call print
0311 01FC02             lxi b,err$select
0314 C32003             jmp goerr1

                goerr:
0317 01EF02             lxi b,err$msg
031A CDCF02             call print
031D 010303             lxi b,err$phys
                goerr1:
0320 CDCF02             call print
0323 F376               di ! hlt

                bde$e$bde$m$hl:
0325 7B955F             mov a,e! sub l! mov e,a
0328 7A9C57             mov a,d! sbb h! mov d,a
032B D005C9             rnc! dcr b! ret
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                        bde$e$bde$p$hl:
032E 7B855F                 mov a,e! add l! mov e,a
0331 7A8C57                 mov a,d! adc h! mov d,a
0334 D004C9                 rnc! inr b! ret

                        sh13bv:
0337 0C                     inr c
                        sh13bv1:
0338 0DC8                   dcr c! rz
033A 298FC33803             dad h! adc a! jmp sh13bv1

                        compare:
033F 1ABEC0                 ldax d! cmp m! rnz
0342 23130DC8               inx h! inx d! dcr c! rz
0346 C33F03                 jmp compare

                        ;
                        ;   local subroutines for bios interface
                        ;

                        move:
                            ; Move data length of length c from source de to
                            ; destination given by hl
0349 0C                     inr c ; in case it is zero
                            move0:
034A 0DC8                       dcr c! rz ; more to move
034C 1A77                       ldax d! mov m,a ; one byte moved
034E 1323                       inx d! inx h ; to next byte
0350 C34A03                     jmp move0

                        selectdisk:
                                        ; Select the disk drive given by register D, and fill
                                        ; the base addresses curtrka - alloca, then fill
                                        ; the values of the disk parameter block
0353 4A                     mov c,d                 ; current disk# to c
                                                    ; lsb of e = 0 if not yet logged - in
0354 CD1B0A                 call seldskf            ; hl filled by call
                                                ; hl = 0000 if error, otherwise disk headers
0357 7CB5C8                 mov a,h! ora l! rz      ; Return with C flag reset if select error
                                                    ; Disk header block address in hl
035A 5E235623               mov e,m! inx h! mov d,m! inx h ; de=.tran
035E 2323                   inx h ! inx h
0360 2290082323             shld curtrka! inx h! inx h ; hl=.currec
0365 2292082323             shld curreca! inx h! inx h ; hl=.buffa
036A 2323                   inx h! inx h
036C 2323                   inx h! inx h
                                                    ; de still contains .tran
036E EB22B808               xchg! shld tranv        ; .tran vector
0372 219808                 lxi h,dpbaddr           ; de= source for move, hl=dest
0375 0E0DCD4903             mvi c,addlist! call move    ; addlist filled
                                                    ; Now fill the disk parameter block
037A 2A9808EB               lhld dpbaddr! xchg      ; de is source
037E 21A508                 lxi h,sectpt            ; hl is destination
0381 0E11CD4903             mvi c,dpblist! call move    ; data filled
                                                    ; Now set single/double map mode
0386 2AAA08                 lhld maxall             ; largest allocation number
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
03B9 7C             mov a,h                         ; 00 indicates < 255
038A 21BD0836FF     lxi h,single! mvi m,true        ; Assume a=00
038F B7CA9503       ora a! jz retselect
                                                    ; high order of maxall not zero, use double dm
0393 3600           mvi m,false
                retselect:
                                                    ; C flag set indicates successful select
0395 37                 stc
0396 C9                 ret

              home:
                                    ; Move to home position, then offset to start of dir
0397 CD180A         call homef
039A AF             xra a                           ; constant zero to accumulator
039B 2A90087723     lhld curtrka! mov m,a! inx h! mov m,a ; curtrk=0000
03A1 2A92087723     lhld curreca! mov m,a! inx h! mov m,a ; currec=0000
03A7 2377           inx h! mov m,a                  ; currec high byte=00

03A9 C9             ret

              pass$arecord:
03AA 21C308         lxi h,arecord
03AD 5E23562346     mov e,m! inx h! mov d,m! inx h! mov b,m
03B2 C9             ret

              rdbuff:
                                    ; Read buffer and check condition
03B3 CDAA03         call pass$arecord
03B6 CD270A         call readf                      ; current drive, track, sector, dma


              diocomp:              ; Check for disk errors
03B9 B7C8           ora a! rz
03BB 4F             mov c,a
03BC FE03DA1703     cpi 3! jc goerr
03C1 0E01C31703     mvi c,1! jmp goerr

              seekdir:
                                    ; Seek the record containing the current dir entry

03C6 2AE308         lhld dcnt                       ; directory counter to hl
03C9 0E02CD2605     mvi c,dskshf! call hlrotr       ; value to hl

03CE 0600EB         mvi b,0! xchg

03D1 21C308         lxi h,arecord
03D4 7323722370     mov m,e! inx h! mov m,d! inx h! mov m,b
03D9 C9             ret

              seek:
                                    ; Seek the track given by arecord (actual record)

03DA 2A900B4E23     lhld curtrka! mov c,m! inx h! mov b,m   ; bc = curtrk
03E0 C5             push b                          ; s0 = curtrk
03E1 2A92085E23     lhld curreca! mov e,m! inx h! mov d,m
03E7 2346           inx h! mov b,m                  ; bde = currec
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
 596  03E9 2AC3083AC5      lhld arecord! lda arecord+2! mov c,    ; chl = arecord
 597                 seek0:
 598  03F0 7D937C9A79      mov a,l! sub e! mov a,h! sbb d! mov a,c! sbb b
 599  03F6 E5              push h                                 ; Save low(arecord)
 600  03F7 D20704          jnc seek1                      ; if arecord >= currec then go to seek1
 601  03FA 2AA508CD25      lhld sectpt! call bde$e$bde$m$hl       ; currec = currec - sectpt
 602  0400 E1E32BE3        pop h! xthl! dcx h! xthl               ; curtrk = curtrk - 1
 603  0404 C3F003          jmp seek0
 604                 seek1:
 605  0407 2AA508CD2E      lhld sectpt! call bde$e$bde$p$hl       ; currec = currec + sectpt
 606  040D E1              pop h                                  ; Restore low(arecord)
 607  040E 7D937C9A79      mov a,l! sub e! mov a,h! sbb d! mov a,c! sbb b
 608  0414 DA1E04          jc seek2                       ; if arecord < currec then go to seek2
 609  0417 E323E3          xthl! inx h! xthl                      ; curtrk = curtrk + 1
 610  041A E5              push h                                 ; save low (arecord)
 611  041B C30704          jmp seek1
 612                 seek2:
 613  041E E3E5            xthl! push h                   ; hl,s0 = curtrk, s1 = low(arecord)
 614  0420 2AA508CD25      lhld sectpt! call bde$e$bde$m$hl       ; currec = currec - sectpt
 615  0426 E1D5C5E5        pop h! push d! push b! push h          ; hl,s0 = curtrk,
 616                                     ; s1 = high(arecord,currec), s2 = low(currec),
 617                                     ; s3 = low(arecord)
 618  042A EB2AB20819      xchg! lhld offset! dad d
 619  042F 444D22CE08      mov b,h! mov c,l! shld track
 620  0434 CD1E0A          call settrkf                           ; call bios settrk routine
 621                                                             ; Store curtrk
 622  0437 D12A900873      pop d! lhld curtrka! mov m,e! inx h! mov m,d
 623                                                             ; Store currec
 624  043E C1D1            pop b! pop d!
 625  0440 2A92087323      lhld curreca! mov m,e! inx h! mov m,d
 626  0446 2370            inx h! mov m,b                         ; currec = bde
 627  0448 C1              pop b                          ; bc = low(arecord), de = low(currec)
 628  0449 79936F          mov a,c! sub e! mov l,a                ; hl = bc - de
 629  044C 789A67          mov a,b! sbb d! mov h,a
 630  044F CD6B04          call shr$physhf
 631  0452 444D            mov b,h! mov c,l
 632
 633  0454 2AB808EB        lhld tranv! xchg                       ; bc=sector$, de=.tran
 634  0458 CD300A          call sectran                           ; hl = tran(sector)
 635  045B 4D44            mov c,l! mov b,h                       ; bc = tran(sector)
 636  045D 22D008          shld sector
 637  0460 CD210A          call setsecf                           ; sector selected
 638  0463 2A8A084D44      lhld curdma! mov c,l! mov b,h! jmp setdmaf
 639
 640                 shr$physhf:
 641  046B 3AB4084FC3      lda physhf! mov c,a! jmp hlrotr
 642
 643
 644                 ;       file control block (fcb) constants
 645
 646  00E5 =       empty  equ     0e5h    ; empty directory entry
 647  0080 =       recsiz equ     128     ; record size
 648  0020 =       fcblen equ     32      ; file control block size
 649  0004 =       dirrec equ     recsiz/fcblen   ; directory fcbs / record
 650  0002 =       dskshf equ     2       ; log2(dirrec)
 651  0003 =       dskmsk equ     dirrec-1
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
0005 =          fcbshf equ      5       ; log2(fcblen)

000C =          extnum equ      12      ; extent number field
001F =          maxext equ      31      ; largest extent number
000D =          ubytes equ      13      ; unfilled bytes field

000F =          namlen equ      15      ; name length
000F =          reccnt equ      15      ; record count field
0010 =          dskmap equ      16      ; disk map field
0020 =          nxtrec equ      fcblen

                ;       utility functions for file access

                dm$position:
                        ; Compute disk map position for vrecord to hl
0472 21A7084E           lxi h,blkshf! mov c,m ; shift count to c
0476 3AC108             lda vrecord ; current virtual record to a
                        dmpos0:
0479 B71F0DC279                 ora a! rar! dcr c! jnz dmpos0
                        ; a = shr(vrecord,blkshf) = vrecord/2**(sect/block)
047F 47                 mov b,a ; Save it for later addition
0480 3E0896             mvi a,8! sub m ; 8-blkshf to accumulator
0483 4F                 mov c,a ; extent shift count in register c
0484 3AC008             lda extval ; extent value ani extmsk
                        dmpos1:
                                ; blkshf = 3,4,5,6,7, c=5,4,3,2,1
                                ; shift is 4,3,2,1,0
0487 0DCA9004                   dcr c! jz dmpos2
048B B717C38704                 ora a! ral! jmp dmpos1
                        dmpos2:
                        ; Arrive here with a = shl(ext and extmsk,7-blkshf)
0490 80                 add b ; Add the previous shr(vrecord,blkshf) value
                        ; a is one of the following values, depending upon alloc
                        ; bks blkshf
                        ; 1k   3     v/8 + extval * 16
                        ; 2k   4     v/16+ extval * 8
                        ; 4k   5     v/32+ extval * 4
                        ; 8k   6     v/64+ extval * 2
                        ; 16k  7     v/128+extval * 1
0491 C9                 ret ; with dm$position in a

                getdma:
0492 2ADF081110         lhld info! lxi d,dskmap! dad d! ret

                getdm:
                        ; Return disk map value from position given by bc
049A CD9204             call getdma
049D 09                 dad b ; Index by a single byte value
049E 3ABD08             lda single ; single byte/map entry?
04A1 B7CAA804           ora a! jz getdmd ; Get disk map single byte
04A5 6E60C9                     mov l,m! mov h,b! ret ; with hl=00bb
                        getdmd:
04A8 09                         dad b ; hl=.fcb(dm+i*2)
                                ; double precision value returned
04A9 7E23666FC9                 mov a,m! inx h! mov h,m! mov l,a! ret
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                        ;lex:
                                ; Compute disk block number from current fcb
04AE CD7204                     call dm$position ; 0...15 in register a
04B1 32BB08                     sta dminx
04B4 4F0600CD9A                 mov c,a! mvi b,0! call getda ; value to hl
04BA 22C3087DB4                 shld arecord! mov a,l! ora h! ret

                        atran:
                                ; Compute actual record address, assuming index called

                        ;       arecord = shl(arecord,blkshf)

04C0 3AA7084F                   lda blkshf! mov c,a
04C4 2AC308AFCD                 lhld arecord! xra a! call shl3bv
04CB 22C30832C5                 shld arecord! sta arecord+2

04D1 22C608                     shld arecord1 ; Save low(arecord)

                        ;       arecord = arecord or (vrecord and blkmsk)

04D4 3AA8084F3A                 lda blkmsk! mov c,a! lda vrecord! ana c
04DC 47                         mov b,a ; Save vrecord & blkmsk in reg b & blk$off
04DD 32B608                     sta blk$off
04E0 21C308B677                 lxi h,arecord! ora m! mov m,a! ret


                        getexta:
                                ; Get current extent field address to hl
04E6 2ADF08110C                 lhld info! lxi d,extnum! dad d ; hl=.fcb(extnum)
04ED C9                         ret

                        getrcnta:
                                ; Get reccnt address to hl
04EE 2ADF08110F                 lhld info! lxi d,reccnt! dad d! ret

                        getfcba:
                                ; Compute reccnt and nxtrec addresses for get/setfcb
04F6 CDEE04EB                   call getrcnta! xchg ; de=.fcb(reccnt)
04FA 21110019                   lxi h,(nxtrec-reccnt)! dad d ; hl=.fcb(nxtrec)
04FE C9                         ret

                        getfcb:
                                ; Set variables from currently addressed fcb
04FF CDF604                     call getfcba ; addresses in de, hl
0502 7E32C108                   mov a,m! sta vrecord ; vrecord=fcb(nxtrec)
0506 EB7E32BF08                 xchg! mov a,m! sta rcount ; rcount=fcb(reccnt)
050B CDE604                     call getexta ; hl=.fcb(extnum)
050E 3AA908                     lda extmsk ; extent mask to a
0511 A6                         a n a m ; fcb(extnum) and extmsk
0512 32C008                     sta extval
0515 C9                         ret

                        setfcb:
                                                        ; Place values back into current fcb
0516 CDF604                     call getfcba            ; addresses to de, hl
0519 0E01                       mvi c,1
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                                
051B 3AC1088177         lda vrecord! add c! mov m,a     ; fcb(nxtrec)=vrecord+seqio
0520 EB3A8F0877         xchg! lda rcount! mov m,a       ; fcb(reccnt)=rcount
0525 C9                 ret
                        
                hlrotr:
                                                        ; hl rotate right by amount c
0526 0C                 inr c                           ; in case zero
0527 0DC8               hlrotr0: dcr c! rz              ; return when zero
                        
0529 7CB71F67           mov a,h! ora a! rar! mov h,a    ; high byte
052D 7D1F6F             mov a,l! rar! mov l,a           ; low byte
0530 C32705             jmp hlrotr0
                        
                hlrotl:
                                                        ; Rotate the mask in hl by amount in c
0533 0C                 inr c                           ; may be zero
0534 0DC8               hlrotl0: dcr c! rz              ; return if zero
                        
0536 29C33405           dad h! jmp hlrotl0
                        
                set$cdisk:
                                                ; Set a "1" value in curdsk position of bc
053A 3ADE08             lda seldsk
053D C5                 push b                          ; Save input parameter
053E 4F                 mov c,a                         ; Ready parameter for shift
053F 210100             lxi h,1                         ; number to shift
0542 CD3305             call hlrotl                     ; hl = mask to integrate
0545 C1                 pop b                           ; original mask
0546 79B56F             mov a,c! ora l! mov l,a
0549 78B467             mov a,b! ora h! mov h,a         ; hl = mask or rol(1,curdsk)
054C C9                 ret
                        
                test$vector:
054D 3ADE08             lda seldsk
0550 4FCD2605           mov c,a! call hlrotr
0554 7DE601C9           mov a,l! ani 1b! ret            ; non zero if curdsk bit on
                        
                getdptra:
                                        ; Compute the address of a directory element at
                                        ; positon dptr in the buffer
                        
0558 2A8C083AC9         lhld buffa! lda dptr
                                                        ; hl = hl + a
055E 856FD0             add l! mov l,a! rnc
                                                        ; overflow to h
0561 24C9               inr h! ret
                        
                clr$ext:
                                        ; fcb ext = fcb ext & 1fh
                        
0563 CDE6047EE6         call getexta! mov a,m! ani 0001$1111b! mov m,a!
056A C9                 ret
                        
                        
                subdh:
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                                        ; Compute hl = de - hl
056B 7B956F7A9C         mov a,e! sub l! mov l,a! mov a,d! sbb h! mov h,a
0571 C9                 ret

                get$buffa:
0572 D5110A0019         push d! lxi d,10! dad d
0577 5E2356             mov e,m! inx h! mov d,m
057A EBD1C9             xchg! pop d! ret


                rddir:
                                        ; Read a directory entry into the directory buffer
057D CDC603             call seek$dir
0580 3AB508B7CA         lda phymsk! ora a! jz rddir1
0587 3E03               mvi a,3
0589 CDE808C39B         call deblock$dir! jmp setdata

                rddir1:
058F CDA105             call setdir                     ; directory dma
0592 228C08CDDA         shld buffa! call seek
0598 CDB303             call rdbuff                     ; directory record loaded

                setdata:
                                        ; Set data dma address
059B 2ADC08C3A7         lhld dmaad! jmp setdma          ; to complete the call

                setdir:
                                        ; Set directory dma address

05A1 2A9E08             lhld dirbcba
05A4 CD7205             call get$buffa

                setdma:
                                        ; hl=.dma address to set (i.e., buffa or dmaad)
05A7 228A08C9           shld curdma! ret

                end$of$dir:
                                        ; Return zero flag if at end of directory, non zero
                                        ; if not at end (end of dir if dcnt = 0ffffh)
05AB 21E308             lxi h,dcnt
05AE 7E                 mov a,m                         ; may be 0ffh
05AF 23BE               inx h! cmp m                    ; low(dcnt) = high(dcnt)?
05B1 C0                 rnz                             ; non zero returned if different
                                                        ; high and low the same, = 0ffh?
05B2 3C                 inr a                           ; 0ffh becomes 00 if so
05B3 C9                 ret

                set$end$dir:
                                        ; Set dcnt to the end of the directory
05B4 21FFFF22E3         lxi h,enddir! shld dcnt! ret


                read$dir:
                                ; Read next directory entry, with c=true if initializing

05BB 2AAC08EB           lhld dirmax! xchg               ; in preparation for subtract
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
876  05BF 2AE3082322      lhld dcnt! inx h! shld dcnt     ; dcnt=dcnt+1
877
878                                                       ; while(dirmax >= dcnt)
879  05C6 CD6B05          call subdh                      ; de-hl
880  05C9 DAB405          jc set$end$dir
881                                          ; not at end of directory, seek next element
882                                          ; initialization flag is in c
883
884  05CC 3AE308E603              lda dcnt! ani dskmsk    ; low(dcnt) and dskmsk
885  05D1 0605                    mvi b,fcbshf            ; to multiply by fcb size
886
887                       read$dir1:
888  05D3 8705C2D305              add a! dcr b! jnz read$dir1
889                                                       ; a = (low(dcnt) and dskmsk) shl fcbshf
890  05D8 32C908                  sta dptr                ; ready for next dir operation
891  05DB B7C0                    ora a! rnz              ; Return if not a new record
892
893  05DD C5                      push b                  ; Save initialization flag c
894  05DE CD7D05                  call rd$dir             ; Read the directory record
895  05E1 C1                      pop b                   ; Recall initialization flag
896  05E2 C9                      ret
897                  compext:
898                                       ; Compare extent# in a with that in c, return nonzero
899                                       ; if they do not match
900  05E3 C5              push b                          ; Save c's original value
901  05E4 F53AA9082F      push psw! lda extmsk! cma! mov b,a
902                                                       ; b has negated form of extent mask
903  05EA 79A04F          mov a,c! ana b! mov c,a         ; low bits removed from c
904  05ED F1A0            pop psw! ana b                  ; low bits removed from a
905  05EF 91E61F          sub c! ani maxext               ; Set flags
906  05F2 C1              pop b                           ; Restore original values
907  05F3 C9              ret
908
909                  get$dir$ext:
910                                       ; Compute directory extent from fcb
911                                       ; Scan fcb disk map backwards
912  05F4 CDF604          call getfcba    ; hl = .fcb(vrecord)
913  05F7 0E10410CC5      mvi c,16! mov b,c! inr c! push b
914                                       ; b=dskmap pos (rel to 0)
915                  get$de0:
916  05FC C1                  pop b
917  05FD 0D                  dcr c
918  05FE AF                  xra a                       ; Compare to zero
919                  get$de1:
920  05FF 2B05                dcx h! dcr b                ; Decr dskmap position
921  0601 BEC20906            cmp m! jnz get$de2          ; fcb(dskmap(b)) <> 0
922  0605 0DC2FF05            dcr c! jnz get$de1
923                                                       ; c = 0 -> all blocks = 0 in fcb disk map
924                  get$de2:
925  0609 7932BB08            mov a,c! sta dminx
926  060D 3ABD08B778          lda single! ora a! mov a,b
927  0612 C21606              jnz get$de3
928  0615 1F                  rar                         ; not single, divide blk idx by 2
929                  get$de3:
930  0616 C5E5                push b! push h              ; Save dskmap position & count
931  0618 6F2600              mov l,a! mvi h,0            ; hl = non-zero blk idx
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                                                ; Compute ext offset from last non-zero
                                                ; block index by shifting blk idx right
                                                ; 7 - blkshf
061B 3AA708573E         lda blkshf! mov d,a! mvi a,7! sub d
0622 4FCD260545         mov c,a! call hlrotr! mov b,l
                                                ; b = ext offset
0627 3AA908B8E1         lda extmsk! cmp b! pop h! jc get$de0
                                        ; Verify computed extent offset <= extmsk
062F CDE6044E           call getexta! mov c,m
0633 2FE61FA1B0         cma! ani maxext! ana c! ora b
                                ; dir ext = (fcb ext & (~ extmsk) & maxext) | ext offset
0638 C1                 pop b                   ; Restore stack
0639 C9                 ret                     ; a = directory extent

                search:
                                        ; Search for directory element of length c at info
063A 2ADF0822E5         lhld info! shld searcha         ; searcha = info
0640 7932E708           mov a,c! sta searchl            ; searchl = c

0644 CDB405             call set$end$dir                ; dcnt = enddir
0647 CD9703             call home                       ; to start at the beginning

                searchn:
                                        ; Search for the next directory element, assuming
                                        ; a previous call on search which sets searcha and
                                        ; searchl

064A 0E00CDBB05         mvi c,false! call read$dir      ; Read next dir element
064F CDAB05CA9B         call end$of$dir! jz lret$eq$ff
                                                ; not end of directory, scan for match
0655 2AE508EB           lhld searcha! xchg              ; de=beginning of user fcb

0659 CD5805                     call getdptra           ; hl = buffa+dptr
065C 3AE7084F                   lda searchl! mov c,a    ; length of search to c
0660 0600                       mvi b,0                 ; b counts up, c counts down

0662 7EFEE5CA4A                 mov a,m! cpi empty! jz searchn

                searchloop:
0668 79B7CA9406                         mov a,c! ora a! jz endsearch
                                                ; Scan next character if not ubytes
066D 78FE0DCA8D                         mov a,b! cpi ubytes! jz searchok
                                                ; not the ubytes field, extent field?
0673 FE0C                               cpi extnum      ; may be extent field
0675 CA8206                             jz searchext    ; Skip to search extent
0678 1A                                 ldax d
0679 96E67F                             sub m! ani 7fh  ; Mask-out flags/extent modulus
067C C24A06                             jnz searchn     ; Skip if not matched
067F C38D06                             jmp searchok    ; matched character
                                searchext:
0682 1A                                 ldax d
                                                ; Attempt an extent # match
0683 C5                                 push b          ; Save counters
0684 4E                                 mov c,m         ; directory character to c
0685 CDE305                             call compext    ; Compare user/dir char
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                                pop b           ; Recall counters
                                ora a           ; Set flag
                                jnz searchn     ; Skip if no match
                        searchok:
                                                ; current character matches
                                inx d! inx h! inr b! dcr c
                                jmp searchloop
                        endsearch:
                                        ; entire name matches, return dir position
                                xra a
                                sta lret        ; lret = 0
                                                ; successful search -
                                                ; return with zero flag reset
                                mov b,a! inr b
                                ret
                        lret$eq$ff:
                                                ; unsuccessful search -
                                                ; return with zero flag set
                                                ; lret,low(aret) = 0ffh
                                mvi a,255 ! mov b,a ! inr b ! jmp sta$ret

                open:
                                ; Search for the directory entry, copy to fcb
                        mvi c,namlen! call search
                        rz                      ; Return with lret=255 if end

                                ; not end of directory, copy fcb information
                open$copy:
                        call getexta ! mov a,m ! push a ; save extent to check for extent
                                                ; folding - move moves entire dir FCB
                        call getdptra! xchg     ; hl = .buff(dptr)
                        lhld info               ; hl=.fcb(0)
                        mvi c,nxtrec            ; length of move operation
                        call move               ; from .buff(dptr) to .fcb(0)

                                ; Note that entire fcb is copied, including indicators

                        call get$dir$ext! mov c,a
                        pop a ! mov m,a         ; restore extent

                                ; hl = .user extent#, c = dir extent#
                                ; above move set fcb(reccnt) to dir(reccnt)
                                ; if fcb ext < dir ext then fcb(reccnt) = fcb(reccnt) ! 128
                                ; if fcb ext = dir ext then fcb(reccnt) = fcb(reccnt)
                                ; if fcb ext > dir ext then fcb(reccnt) = 0

                set$rc:                         ; hl=.fcb(ext), c=dirext
                        mvi b,0
                        xchg! lxi h,(reccnt-extnum)! dad d
                        ldax d! sub c! jz set$rc2
                        mov a,b! jnc set$rc1
                        mvi a,128! mov b,m

                 set$rc1:
                        mov m,a! mov a,b! sta actual$rc! ret
                set$rc2:
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
1044  06D8 32BC08              sta actual$rc
1045  06DB 7EB7C0              mov a,m! ora a! rnz     ; ret if rc "= 0
1046  06DE 3ABB08B7C8          lda dminx! ora a! rz    ; ret if no blks in fcb
1047  06E3 3AE208FE0F          lda fx! cpi 15! rz      ; ret if fx = 15
1048  06E9 3680                mvi m,128               ; rc = 128
1049  06EB C9                  ret
1050
1051              restore$rc:
1052                                   ; hl = .fcb(extnum)
1053                                   ; if actual$rc "= 0 then rcount = actual$rc
1054  06EC E5             push h
1055  06ED 3ABC08B7CA     lda actual$rc! ora a! jz restore$rc1
1056  06F4 11030019       lxi d,(reccnt-extnum)! dad d
1057  06F8 77AF32BC08     mov m,a! xra a! sta actual$rc
1058
1059              restore$rc1:
1060  06FD E1C9           pop h! ret
1061
1062              open$reel:
1063                                   ; Close the current extent, and open the next one
1064                                   ; if possible.
1065
1066  06FF CDE604         call getexta
1067  0702 7E4F           mov a,m! mov c,a
1068  0704 0CCDE305       inr c! call compext
1069  0708 CA2107         jz open$reel3
1070
1071  070B 3E1FA177       mvi a,maxext! ana c! mov m,a        ; Incr extent field
1072  070F 0E0FCD3A06     mvi c,namlen! call search           ; Next extent found?
1073                                                          ; not end of file, open
1074  0714 CDA806         call open$copy
1075
1076                open$reel2:
1077  0717 CDFF04               call getfcb                   ; Set parameters
1078  071A AF32C108C3           xra a! sta vrecord! jmp sta$ret ; lret = 0
1079                open$reel3:
1080  0721 34                   inr m                         ; fcb(ex) = fcb(ex) + 1
1081  0722 CDF4054F             call get$dir$ext! mov c,a
1082                                                          ; Is new extent beyond dir$ext?
1083  0726 BED22E07             cmp m! jnc open$reel4         ; no
1084  072A 35                   dcr m                         ; fcb(ex) = fcb(ex) - 1
1085  072B C3E902               jmp set$lret1
1086                open$reel4:
1087  072E CDEC06               call restore$rc
1088  0731 CDBF06C317           call set$rc! jmp open$reel2
1089
1090              seqdiskread:
1091                                   ; Sequential disk read operation
1092                                   ; Read the next record from the current fcb
1093
1094  0737 CDFF04         call getfcb                         ; sets parameters for the read
1095
1096  073A 3AC10821BF     lda vrecord! lxi h,rcount! cmp m     ; vrecord-rcount
1097                                                          ; Skip if rcount > vrecord
1098  0741 DA5307         jc recordok
1099
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                                        ; not enough records in the extent
                                        ; record count must be 128 to continue
0744 FE80               cpi 128                 ; vrecord = 128?
0746 C2E902             jnz setlret1            ; Skip if vrecord<>128
0749 CDFF06             call open$reel          ; Go to next extent if so
                                                ; Check for open ok
074C 3A0903B7C2         lda lret! ora a! jnz setlret1   ; Stop at eof

            recordok:
                                ; Arrive with fcb addressing a record to read

0753 CDAE04             call index              ; Z flag set if arecord = 0

0756 CAE902             jz setlret1             ; Reading unwritten data

                                                ; Record has been allocated
0759 CDC004             call atran              ; arecord now a disk address

075C 3AB508B7           lda physk! ora a        ; if not 128 byte sectors
0760 C2D208             jnz read$deblock        ; go to deblock

0763 CD9B05             call setdata            ; Set curdma = dmaad
0766 CDDA03             call seek               ; Set up for read
0769 CDB303             call rdbuff             ; Read into (curdma)
076C C31605             jmp setfcb              ; Update FCB

          curselect:
076F 3ADE083CCA     lda seldsk! inr a! jz sel$error
0776 3D21C208BE     dcr a! lxi h,curdsk! cmp m! rz

                                        ; Skip if seldsk = curdsk, fall into select
          select:
                                ; Select disk info for subsequent input or output ops
077C 77             mov m,a                     ; curdsk = seldsk

077D 57             mov d,a             ; Save seldsk in register D for selectdisk call
077E 2A8808CD4D     lhld dlog! call test$vector ; test$vector does not modify DE
0784 5FD5           mov e,a! push d             ; Send to seldsk, save for test below
0786 CD5303E1       call selectdisk! pop h      ; Recall dlog vector
078A D20B03         jnc sel$error               ; returns with C flag set if select ok
                                                ; Is the disk logged in?
078D 2D             dcr l                       ; reg l = 1 if so
078E C8             rz                          ; yes - drive previously logged in

078F 2A8808D44      lhld dlog! mov c,l! mov b,h ; call ready
0794 CD3A052288     call set$cdisk! shld dlog   ; dlog=set$cdisk(dlog)
079A C9             ret

          set$seldsk:
079B 3ABA0832DE     lda linfo! sta seldsk! ret

          reselectx:
07A2 AF32C808C3     xra a! sta high$ext! jmp reselect1
          reselect:
                                ; Check current fcb to see if reselection necessary
07A9 3E80473D4F     mvi a,80h! mov b,a! dcr a! mov c,a   ; b = 80h, c = 7fh
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
07AE 2ADF081107       lhld info! lxi d,7! xchg! dad d
07B6 7EA0             mov a,m! ana b
                                                    ; fcb(7) = fcb(7) & 7fh
07B8 7EA177           mov a,m! ana c! mov m,a
                                                    ; high$ext = 80h & fcb(8)
07BB 237EA032C8       inx h! mov a,m! ana b! sta high$ext
                                                    ; fcb(8) = fcb(8) & 7fh
07C1 7EA177           mov a,m! ana c! mov m,a
                                                    ; fcb(ext) = fcb(ext) & 1fh
07C4 CD6305           call clr$ext

                      ; if fcb(rc) & 80h
                      ;    then fcb(rc) = 80h, actual$rc = fcb(rc) & 7fh
                      ;    else actual$rc = 0

07C7 CDEE047EA0       call getrcnta! mov a,m! ana b! jz reselect1
07CF 7EA170           mov a,m! ana c! mov m,b

                reselect1:
07D2 328C08           sta actual$rc

07D5 210000           lxi h,0
07D8 22CA08           shld fcbdsk                   ; fcbdsk = 0
07DB 3EFF32E108       mvi a,true! sta resel         ; Mark possible reselect
07E0 2ADF087E         lhld info! mov a,m            ; drive select code
07E4 E61F             ani 1$1111b                   ; non zero is auto drive select
07E6 3D               dcr a                 ; Drive code normalized to 0,,30, or 255
07E7 32BA08           sta linfo                     ; Save drive code
07EA FEFFCAF607       cpi 0ffh! jz noselect
                                            ; auto select function, seldsk saved above
07EF 7E32CA08         mov a,m! sta fcbdsk           ; Save drive code
07F3 CD9B07           call set$seldsk

                 noselect:
07F6 CD6F07                 call curselect
07F9 3E002ADF08             mvi a,0 ! lhld info ! mov m,a
07FF C9                     ret

                ;
                ;     individual function handlers
                ;

02E8 =          func12 equ func$ret

                func13:

                                    ; Reset disk system - initialize to disk 0
0800 2100002288       lxi h,0! shld dlog

0806 AF32DE08         xra a! sta seldsk
080A 3D32C208         dcr a! sta curdsk

080E 21800022DC       lxi h,tbuff! shld dmaad       ; dmaad = tbuff
0814 C39B05            jmp setdata                  ; to data dma address

                func14:
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
1212                                    ; Select disk info
1213  0817 CD9B07       call set$seldsk                 ; seldsk = linfo
1214  081A C36F07       jmp curselect
1215
1216            func15:
1217                                    ; Open file
1218  081D CDA207       call reselectx
1219  0820 CDA206CD27   call open! call openx          ; returns if unsuccessful, a = 0
1220  0826 C9           ret
1221
1222            openx:
1223  0827 CDAB05C8     call end$of$dir! rz
1224  082B CDF6047E3C   call getfcba! mov a,m! inr a! jnz openxa
1225  0833 1B1B1A77     dcx d! dcx d! ldax d! mov m,a
1226            openxa:
1227                                                    ; open successful
1228  0837 E1           pop h                           ; Discard return address
1229  0838 0E40         mvi c,0100$0000b
1230  083A C9           ret
1231
1232  02E8 =      func16 equ func$ret
1233
1234  02E8 =      func17 equ func$ret
1235
1236  02E8 =      func18 equ func$ret
1237
1238  02E8 =      func19 equ func$ret
1239
1240            func20:
1241                                            ; Read a file
1242  083B CDA907       call reselect
1243  083E C33707       jmp seqdiskread
1244
1245  02E8 =      func21 equ func$ret
1246
1247  02E8 =      func22 equ func$ret
1248
1249  02E8 =      func23 equ func$ret
1250
1251  02E8 =      func24 equ func$ret
1252
1253  0841 3ADE08C3E5func25: lda seldsk ! jmp sta$ret
1254
1255  0847 EB22DC08  func26:      xchg ! shld dmaad
1256  084B C39B05       jmp setdata
1257
1258  02E8 =      func27 equ func$ret
1259
1260  02E8 =      func28:     equ func$ret
1261
1262  02E8 =      func29 equ func$ret
1263
1264  02E8 =      func30 equ func$ret
1265
1266  02E8 =      func31 equ func$ret
1267
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
1268  02E8 =        func32 equ func$ret
1269
1270  02E8 =        func33 equ func$ret
1271
1272  02E8 =        func34 equ func$ret
1273
1274  02E8 =        func35 equ func$ret
1275
1276  02E8 =        func36 equ func$ret
1277
1278  02E8 =        func37 equ func$ret
1279
1280  02E8 =        func38 equ func$ret
1281
1282  02E8 =        func39 equ func$ret
1283
1284  02E8 =        func40  equ func$ret
1285
1286  02E8 =        func42 equ func$ret
1287
1288  02E8 =        func43 equ func$ret
1289
1290  02E8 =        func44 equ func$ret
1291
1292  02E8 =        func45 equ func$ret
1293
1294  02E8 =        func46 equ func$ret
1295
1296  02E8 =        func47 equ    func$ret
1297
1298  02E8 =        func48 equ func$ret
1299
1300  02E8 =        func49 equ    func$ret
1301
1302  02E8 =        func50 equ    func$ret
1303
1304  02E8 =        func100       equ func$ret
1305
1306  02E8 =        func101       equ func$ret
1307
1308  02E8 =        func102       equ func$ret
1309
1310  02E8 =        func103       equ func$ret
1311
1312  02E8 =        func104       equ func$ret
1313
1314  02E8 =        func105       equ func$ret
1315
1316  02E8 =        func106       equ func$ret
1317
1318  02E8 =        func107       equ func$ret
1319
1320  02E8 =        func108       equ func$ret
1321
1322  02E8 =        func109       equ func$ret
1323
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
1324
1325                goback:
1326                                        ; Arrive here at end of processing to return to user
1327  084E 3AE208FE0F       lda fx! cpi 15! jc retmon
1328  0856 3ABE0832DE       lda olddsk! sta seldsk                  ; Restore seldsk
1329  085C 3AE108B7CA       lda resel! ora a! jz retmon
1330
1331  0863 2ADF083600       lhld info! mvi m,0 ; fcb(0)=0
1332  0868 3ACA08B7CA       lda fcbdsk! ora a! jz goback1
1333                                                                ; Restore fcb(0)
1334  086F 77               mov m,a                                 ; fcb(0)=fcbdsk
1335                    goback1:
1336                                                                ; fcb(8) = fcb(8) ! high$ext
1337  0870 233AC808B6       inx h! lda high$ext! ora m! mov m,a
1338                                                                ; fcb(rc) = fcb(rc) ! actual$rc
1339  0876 CDEE043ABC       call getrcnta! lda actual$rc! ora m! mov m,a
1340                                                                ; return from the disk monitor
1341                retmon:
1342  087E 2A4002F9         lhld entsp! sphl
1343  0882 2A09037D44       lhld aret! mov a,l! mov b,h
1344  0887 C9               ret
1345                ;
1346                ;       data areas
1347                ;
1348  0888 0000     dlog:   dw      0       ; logged-in disks
1349  088A          curdma  ds      word    ; current dma address
1350  088C          buffa:  ds      word    ; pointer to directory dma address
1351
1352                ;
1353                ;       curtrka - alloca are set upon disk select
1354                ;       (data must be adjacent, do not insert variables)
1355                ;       (address of translate vector, not used)
1356  088E          cdrmaxa:ds      word    ; pointer to cur dir max value (2 bytes)
1357  0890          curtrka:ds      word    ; current track address (2)
1358  0892          curreca:ds      word    ; current record address (3)
1359  0894          drvlbla:ds      word    ; current drive label byte address (1)
1360  0896          lsn$add:ds      word    ; login sequence # address (1)
1361                                        ; +1 -> bios media change flag (1)
1362  0898          dpbaddr:ds      word    ; current disk parameter block address
1363  089A          checka:         ds      word    ; current checksum vector address
1364  089C          alloca:         ds      word    ; current allocation vector address
1365  089E          dirbcba:ds      word    ; dir bcb list head
1366  08A0          dtabcba:ds      word    ; data bcb list head
1367                hash$tbla:
1368  08A2                  ds      word
1369  08A4                  ds      byte
1370
1371  000D =        addlist         equ     $-dpbaddr       ; address list size
1372
1373                ;
1374                ;               buffer control block format
1375                ;
1376                ; bcb format : drv(1) !! rec(3) !! pend(1) !! sequence(1) !!
1377                ;               0       1       4       5
1378                ;
1379                ;               track(2) !! sector(2) !! buffer$add(2) !!
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1380                    ;               6           8           10
1381                    ;
1382                    ;               link(2)
1383                    ;               12
1384                    ;
1385
1386                    ;       sectpt - offset obtained from disk parm block at dpbaddr
1387                    ;       (data must be adjacent, do not insert variables)
1388    08A5            sectpt: ds      word    ; sectors per track
1389    08A7            blkshf: ds      byte    ; block shift factor
1390    08A8            blkmsk: ds      byte    ; block mask
1391    08A9            extmsk: ds      byte    ; extent mask
1392    08AA            maxall: ds      word    ; maximum allocation number
1393    08AC            dirmax: ds      word    ; largest directory number
1394    08AE            dirblk: ds      word    ; reserved allocation bits for directory
1395    08B0            chksiz: ds      word    ; size of checksum vector
1396    08B2            offset: ds      word    ; offset tracks at beginning
1397    08B4            physhf: ds      byte    ; physical record shift
1398    08B5            phymsk: ds      byte    ; physical record mask
1399    0011 =          dpblist equ     $-sectpt        ; size of area
1400                    ;
1401                    ;       local variables
1402                    ;
1403    08B6            blk$off: ds     byte    ; record offset within block
1404    08B7            dir$cnt: ds     byte    ; direct i/o count
1405
1406    08B8            tranv: ds       word    ; address of translate vector
1407    08BA            linfo: ds       byte    ; low(info)
1408    08BB            dminx: ds       byte    ; local for diskwrite
1409
1410                    actual$rc:
1411    08BC                    ds      byte    ; directory ext record count
1412
1413    08BD            single: ds      byte    ; set true if single byte allocation map
1414
1415
1416    08BE            olddsk: ds      byte    ; disk on entry to bdos
1417    08BF            rcount: ds      byte    ; record count in current fcb
1418    08C0            extval: ds      byte    ; extent number and extmsk
1419
1420    08C1            vrecord:ds      byte    ; current virtual record
1421
1422                    curdsk:
1423
1424    08C2 FF         adrive: db      0ffh    ; current disk
1425    08C3            arecord:ds      word    ; current actual record
1426    08C5                    ds      byte
1427
1428    08C6            arecord1: ds    word    ; current actual block# * blkmsk
1429
1430                    ;******** following variable order critical ******************
1431
1432    08C8            high$ext: ds    byte    ; fcb high ext bits
1433                    ;xfcb$read$only: ds     byte
1434
1435                    ;       local variables for directory access
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

```
08C9          dptr:  ds      byte    ; directory pointer 0,1,2,3

                ;
                ;       local variables initialized by bdos at entry
                ;
08CA          fcbdsk:                ds      byte    ; disk named in fcb

08CB          phy$off:       ds      byte
08CC          curbcba:       ds      word

08CE          track:         ds      word
08D0          sector:                ds      word

                read$deblock:
08D2 3E01CDEE08         mvi a,1! call deblock$dta
08D7 C31605             jmp setfcb

08DA 00       column          db      0
08DB 24       outdelim:       db      '$'

08DC 8000     dmaad:          dw      0080h
08DE 00       seldsk:                 db      0
08DF 0000     info:           dw      0
08E1 00       resel:          db      0
08E2 00       fx:             db      0
08E3 0000     dcnt:           dw      0
08E5 0000     searcha:        dw      0
08E7 00       searchl:        db      0

                ;       ***************************
                ;       Blocking/Deblocking Module
                ;       ***************************

                deblock$dir:

08E8 2A9E08             lhld dirbcba

08EB C3F108             jmp deblock

                deblock$dta:
08EE 2AA008             lhld dtabcba

                deblock:

                        ; BDOS Blocking/Deblocking routine
                        ; a = 1 -> read command
                        ; a = 2 -> write command
                        ; a = 3 -> locate command
                        ; a = 4 -> flush command
                        ; a = 5 -> directory update

08F1 F5                 push a                          ; Save z flag and deblock fx

                                                        ; phy$off = low(arecord) & physmsk
                                                        ; low(arecord) = low(arecord) & ^physmsk
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
1492  08F2 CD5409          call deblock8
1493  08F5 3AC3085FA0      lda arecord! mov e,a! ana b! sta phy$off
1494  08FD 7BA132C308      mov a,e! ana c! sta arecord
1495
1496  0902 22CC08CD72      shld curbcba! call getbuffa! shld curdma
1497
1498  090B CD5B09          call deblock9
1499                                                ; Is command flush?
1500  090E F1F5FE04        pop a! push a! cpi 4
1501  0912 D21C09          jnc deblock1             ; yes
1502                                                ; Is referenced physical record
1503                                                ;already in buffer?
1504  0915 CD3F03CA2C      call compare! jz deblock45  ; yes
1505  091B AF              xra a
1506                deblock1:
1507  091C CD6409          call deblock10
1508                                                ; Read physical record buffer
1509  091F 3E02CD6C09      mvi a,2! call deblock$io
1510
1511  0924 CD5B09          call deblock9            ; phypfx = adrive !! arecord
1512  0927 CD49033600      call move! mvi m,0       ; zero pending flag
1513
1514              deblock45:
1515                                                ; recadd = phybuffa + phy$off*80h
1516  092C 3ACB083C11      lda phy$off! inr a! lxi d,80h! lxi h,0ff80h
1517              deblock5:
1518  0936 193DC23609      dad d! dcr a! jnz deblock5
1519  093B EB2A8A0819      xchg! lhld curdma! dad d
1520                                                ; If deblock command = locate
1521                                                ; then buffa = recadd; return
1522  0940 F1FE03C24A      pop a! cpi 3! jnz deblock6
1523  0946 228C08C9        shld buffa! ret
1524              deblock6:
1525  094A EB2ADC0801      xchg! lhld dmaad! lxi b,80h
1526                                                ; If deblock command = read
1527  0951 C34B0A          jmp move$tpa             ; then move to dma
1528
1529              deblock8:
1530  0954 3AB508472F      lda phymsk! mov b,a! cma! mov c,a! ret
1531
1532              deblock9:
1533  095B 2ACC0811C2      lhld curbcba! lxi d,adrive! mvi c,4! ret
1534
1535              deblock10:
1536  0964 110400          lxi d,4
1537              deblock11:
1538  0967 2ACC0819C9      lhld curbcba! dad d! ret
1539
1540              deblock$io:
1541                                                ; a = 0 -> seek only
1542                                                ; a = 1 -> write
1543                                                ; a = 2 -> read
1544  096C F5CDDA03        push a! call seek
1545  0970 F13D            pop a! dcr a
1546  0972 F4B303          cp rdbuff
1547                                                ; Move track & sector to bcb
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
1548  0975 CD64092323       call deblock10! inx h! inx h
1549  097A 11CE080E04       lxi d,track! mvi c,4! jmp move
1550
1551  09FF                  org     base+(($-base)+255) and 0ff00h)-1
1552  09FF 00               db      0
1553
1554                  ; Bios equates
1555
1556  0A00 =          bios$pg               equ     $
1557
1558  0A00 =          bootf           equ     bios$pg+00      ; 00. cold boot
1559  0A0C =          conoutf                 equ     bios$pg+12      ; 04. console output function
1560  0A18 =          homef           equ     bios$pg+24      ; 08. disk home function
1561  0A1B =          seldskf                 equ     bios$pg+27      ; 09. select disk function
1562  0A1E =          settrkf                 equ     bios$pg+30      ; 10. set track function
1563  0A21 =          setsecf                 equ     bios$pg+33      ; 11. set sector function
1564  0A24 =          setdmaf                 equ     bios$pg+36      ; 12. set dma function
1565  0A30 =          sectran                 equ     bios$pg+48      ; 16. sector translate
1566  0A4B =          movef           equ     bios$pg+75      ; 25. memory move function
1567  0A27 =          readf           equ     bios$pg+39      ; 13. read disk function
1568  0A4B =          move$out        equ     movef
1569  0A4B =          move$tpa        equ     movef
1570
1571  0A00                            end
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____ ____

ABASE FF00 13# 18 19 32 33 34 35
ACTUALRC 08BC 1042 1044 1055 1057 1175 1339 1410#
ADDLIST 000D 533 1371#
ADRIVE 08C2 1424# 1533
ALLOCA 089C 1364#
ARECORD 08C3 561 585 596 596 713 721 722 722 731 1425#
1493 1494
ARECORD1 08C6 724 1428#
ARET 0309 258 398 439# 440 1343
ATRAN 04C0 715# 1116
BANKLEN 0184 94 206#
BANKTOP 0183 98 204#
BASE 0000 12# 13 1551 1551
BDEEBDEMHL 0325 479# 601 614
BDEEBDEPHL 032E 484# 605
BDOS 018D 45 49 53 78 149 158 167 247#
BDOSE 018D 248#
BDOSE2 01A5 251 256#
BDOSJMP 01CF 269 275#
BIOSPG 0A00 1556# 1558 1559 1560 1561 1562 1563 1564 1565 1566
1567
BLKMSK 08A8 728 1390#
BLKOFF 08B6 730 1403#
BLKSHF 08A7 667 720 935 1389#
BNKLEN FF83 35#
BNKTOP FF82 34#
BOOTF 0A00 42 1558#
BREAK 00A9 111 171#
BUFF FF80 19# 58 63 77
BUFFA 088C 806 839 1350# 1523
BYTE 0001 455# 1369 1389 1390 1391 1397 1398 1403 1404 1407
1408 1411 1413 1416 1417 1418 1420 1426 1432 1436
1441 1443
CDRMAXA 088E 1356#
CHECKA 089A 1363#
CHKSIZ 08B0 1395#
CLOOP 0034 66# 72
CLREXT 0563 812# 1165
COLUMN 08DA 345 373 1453#
COMLEN FF81 33#
COMPARE 033F 495# 498 1504
COMPCOL 02EE 336 419#
COMPEXT 05E3 897# 987 1068
COMPOUT 029E 336 342#
COMTOP FF80 32#
CONOUT 0292 333# 369 372
CONOUTF 0A0C 340 1559#
CPMFCB 00AB 52 157 175#
CR 000D 15# 180 182 185 187 190 193 195 [illegible]# 429
CTLH 0008 237# 354
CURBCBA 08CC 1444# 1496 1533 1538
CURDMA 088A 638 854 1349# 1496 1519
CURDSK 08C2 1128 1206 1422#
CURRECA 0892 527 555 594 625 1358#
CURSELECT 076F 1126# 1190 1214
CURTRKA 0890 526 554 592 622 1357#

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
DCNT            08E3   580   859   869   876   876   884  1461#
DEBLOCK         08F1  1474  1479#
DEBLOCK1        091C  1501  1506#
DEBLOCK10       0964  1507  1535# 1548
DEBLOCK11       0967  1537#
DEBLOCK45       092C  1504  1514#
DEBLOCK5        0936  1517# 1518
DEBLOCK6        094A  1522  1524#
DEBLOCK8        0954  1492  1529#
DEBLOCK9        095B  1498  1511  1532#
DEBLOCKDIR      08E8   835  1470#
DEBLOCKDTA      08EE  1450  1476#
DEBLOCKIO       096C  1509  1540#
DIOCOMP         03B9   571#
DIRBCBA         089E   849  1365# 1472
DIRBLK          08AE  1394#
DIRCNT          08B7   253  1404#
DIRMAX          08AC   875  1393#
DIRREC          0004   649#  651
DISKF           000C   289#
DLOG            0888  1136  1144  1145  1203  1348#
DMAAD           08DC   844  1208  1255  1456# 1525
DMINX           08BB   711   925  1046  1408#
DMPOS0          0479   669#  670
DMPOS1          0487   676#  680
DMPOS2          0490   679   681#
DMPOSITION      0472   665#  710
DPBADDR         0898   532   535  1362# 1371
DPBLIST         0011   537  1399#
DPTR            08C9   806   890  1436#
DRVLBLA         0894  1359#
DSKMAP          0010   660#  694
DSKMSK          0003   651#  884
DSKSHF          0002   581   650#
DTABCBA         08A0  1366# 1477
EMPTY           00E5   646#  969
ENDDIR          FFFF   454#  869
ENDOFDIR        05AB   856#  961  1223
ENDSEARCH       0694   972   995#
ENTSP           0240   261   303# 1342
ERRMSG          02EF   429#  466   472
ERROR           00A2    56   165#
ERRPHYS         0303   431#  474
ERRSELECT       02FC   430#  468
EXECUTE         005F    96   103#
EXECUTESYS      006F   107   112#
EXTMSK          08A9   755   901   938  1391#
EXTNUM          000C   654#  736   976  1036  1056
EXTVAL          08C0   675   757  1418#
FALSE           0000   453#  544   960
FCB             FF5C    18#  104
FCBDSK          08CA  1178  1186  1332  1441#
FCBLEN          0020   648#  649   661
FCBSHF          0005   652#  885
FUNC1           02E8   286   406#
FUNC10          02E8   288   412#
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

FUNC100	02E8	130[illegible]	
FUNC101	02E8	1306#	
FUNC102	02E8	1308#	
FUNC103	02E8	1310#	
FUNC104	02E8	1312#	
FUNC105	02E8	1314#	
FUNC106	02E8	1316#	
FUNC107	02E8	1318#	
FUNC108	02E8	1320#	
FUNC109	02E8	1322#	
FUNC11	02E8	28[illegible]	413#
FUNC12	02E8	290	1198#
FUNC13	0800	290	1200#
FUNC14	0817	290	1211#
FUNC15	081D	290	1216#
FUNC16	02E8	291	1232#
FUNC17	02E8	291	1234#
FUNC18	02E8	291	1236#
FUNC19	02E8	291	1238#
FUNC2	02BB	286	387#
FUNC20	083B	292	1240#
FUNC21	02E8	292	1245#
FUNC22	02E8	292	1247#
FUNC23	02E8	292	1249#
FUNC24	02E8	293	1251#
FUNC25	0841	293	1253#
FUNC26	0847	293	1255#
FUNC27	02E8	293	1258#
FUNC28	02E8	294	1260#
FUNC29	02E8	294	1262#
FUNC3	02E8	286	408#
FUNC30	02E8	294	1264#
FUNC31	02E8	294	1266#
FUNC32	02E8	295	1268#
FUNC33	02E8	295	1270#
FUNC34	02E8	295	1272#
FUNC35	02E8	295	1274#
FUNC36	02E8	296	1276#
FUNC37	02E8	296	1278#
FUNC38	02E8	296	1280#
FUNC39	02E8	296	1282#
FUNC40	02E8	297	1284#
FUNC42	02E8	297	1286#
FUNC43	02E8	297	1288#
FUNC44	02E8	298	1290#
FUNC45	02E8	298	1292#
FUNC46	02E8	298	1294#
FUNC47	02E8	298	1296#
FUNC48	02E8	299	1298[illegible]
FUNC49	02E8	299	1300#
FUNC50	02E8	299	1302#
FUNC6	02E8	287	410#
FUNC9	02DF	288	390#
FUNCRET	02E8	286	287 287 287 288 399# 406 408 410 412
		413	1198 1232 1234 1236 1238 1245 1247 1249 1251
		1258	1260 1262 1264 1266 1268 1270 1272 1274 1276

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                    1278  1280  1282  1284  1286  1288  1290  1292  1294  1296
                    1298  1300  1302  1304  1306  1308  1310  1312  1314  1316
                    1318  1320  1322
FUNCTAB     01DC    269   285#  289   300
FX          08E2    252   1047  1327  1460#
GETBUFFA    0572    824#  850   1496
GETDE0      05FC    915#  938
GETDE1      05FF    919#  922
GETDE2      0609    921   924#
GETDE3      0616    927   929#
GETDIREXT   05F4    909#  1025  1081
GETDM       049A    696#  712
GETDMA      0492    693#  698
GETDMD      04A8    701   703#
GETDPTRA    0558    802#  965   1018
GETEXTA     04E6    734#  754   815   940   1016  1066
GETFCB      04FF    749#  1077  1094
GETFCBA     04F6    743#  751   762   912   1224
GETRCNTA    04EE    739#  745   1171  1339
GOBACK      084E    265   1325#
GOBACK1     0870    1332  1335#
GOERR       0317    471#  574   575
GOERR1      0320    469   475#
HASHTBLA    08A2    1367#
HIGHEXT     08C8    1152  1161  1337  1432#
HIGHFXS     01CA    267   272#
HLROTL      0533    778#  791
HLROTL0     0534    781#  783
HLROTR      0526    581   641   769#  799   936
HLROTR0     0527    772#  776
HOME        0397    550#  953
HOMEF       0A18    552   1560#
INDEX       04AE    708#  1111
INFO        08DF    249   279   694   736   741   949   1019  1156  1180  1191
                    1331  1458#
LF          000A    16#   180   182   185   187   191   191   191   191   191
                    191   191   191   191   191   191   191   192   192   192
                    192   192   192   192   192   192   192   192   192   193
                    195   236#  361   429
LINFO       08BA    257   1149  1183  1407#
LOAD        0073    90    99    122#
LOOP        007A    129#  141
LRET        0309    440#  998   1106
LRETE0FF    069B    273   961   1003#
LSNADD      0896    1360#
LSTACK      0292    263   312#
MAXALL      08AA    539   1392#
MAXEXT      001F    655#  905   941   1071
MD          0024    427#  429   430   431
MEMTOP      0181    64    89    200#
MOVE        0349    504#  533   537   1021  1512  1549
MOVE0       034A    508#  512
MOVEF       0A4B    1566# 1568  1569
MOVEOUT     0A4B    1568#
MOVETPA     0A4B    1527  1569#
NAMLEN      000F    658#  1011  1072
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
NFUNCS          0032  267   300#
NOSELECT        07F6  1184  1189#
NOTBACKSP       02B5  354   359#
NXTREC          0020  661#  746  1020
OFFSET          08B2  618   1396#
OLDDSK          08BE  254   1328  1416#
OPEN            06A2  1009# 1219
OPENCOPY        06A8  1015# 1074
OPENERR         00CF  55    179#
OPENFUNC        000F  26#   51
OPENREEL        06FF  1062# 1104
OPENREEL2       0717  1076# 1088
OPENREEL3       0721  1069  1079#
OPENREEL4       072E  1083  1086#
OPENX           0827  1219  1222#
OPENXA          0837  1224  1226#
OSENTRY         FF84  36#
OSENTRYADR      0185  113   208#
OUTDELIM        08DB  379   1454#
PASSARECORD     03AA  560#  567
PHYMSK          08B5  833   1118  1398# 1530
PHYOFF          08CB  1443# 1493  1516
PHYSHF          08B4  641   1397#
PRINT           02CF  377#  384   394   467   473   476
PRINTBUF        0009  25#   47    76    166
RCOUNT          08BF  753   766   1096  1417#
RDBUFF          03B3  565#  840   1123  1546
RDDIR           057D  830#  894
RDDIR1          058F  833   837#
READDEBLOCK     08D2  1119  1449#
READDIR         05BB  872#  960
READDIR1        05D3  887#  888
READERR         00FA  160   184#
READF           0A27  568   1567#
READFUNC        0014  27#   156
READPROC        0095  61    74    137   155#
RECCNT          000F  659#  741   746   1036  1056
RECORDOK        0753  1098  1108#
RECSIZ          0080  647#  649
RESEL           08E1  259   1179  1329  1459#
RESELECT        07A9  1153# 1242
RESELECT1       07D2  1152  1171  1174#
RESELECTX       07A2  1151# 1218
RESETSYS        000D  24#   44
RESLEN          0182  87    202#
RESTORERC       06EC  1051# 1087
RESTORERC1      06FD  1055  1059#
RETMON          087E  1327  1329  1341#
RETSELECT       0395  542   545#
RUBOUT          007F  233#  346
SEARCH          063A  947#  1011  1072
SEARCHA         08E5  949   963   1462#
SEARCHEXT       0682  977   982#
SEARCHL         08E7  950   966   1463#
SEARCHLOOP      0668  971#  994
SEARCHN         064A  955#  969   980   990
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
SEARCHOK        068D  974  981  991#
SECTOR          08D0  636  1447#
SECTPT          08A5  536  601  605  614  1388# 1399
SECTRAN         0A30  634  1565#
SEEK            03DA  589#  839  1122  1544
SEEK0           03F0  597#  603
SEEK1           0407  600  604#  611
SEEK2           041E  608  612#
SEEKDIR         03C6  577#  832
SELDSK          08DE  254  787  798  1127  1149  1205  1253  1328  1457#
SELDSKF         0A1B  520  1561#
SELECT          077C  1131#
SELECTDISK      0353  514#  1138
SELERROR        030B  464#  1127  1139
SEQDISKREAD     0737  1090#  1243
SERIAL          0187  241#
SETCDISK        053A  785#  1145
SETDATA         059B  835  842#  1121  1209  1256
SETDIR          05A1  838  846#
SETDMA          05A7  844  852#
SETDMAF         0A24  638  1564#
SETDMAFUNC      001A  28#  148
SETDMAPROC      008F  59  136  147#
SETENDDIR       05B4  867#  880  952
SETFCB          0516  760#  1124  1451
SETLRET1        02E9  402#  1085  1103  1106  1113
SETRC           06BF  1034#  1088
SETRC1          06D2  1038  1041#
SETRC2          06D8  1037  1043#
SETSECF         0A21  637  1563#
SETSELDSK       079B  1148#  1187  1213
SETTRKF         0A1E  620  1562#
SHL3BV          0337  489#  721
SHL3BV1         0338  491#  493
SHRPHYSHF       046B  630  640#
SIGNON          0125  48  189#
SINGLE          08BD  541  700  926  1413#
STACKBOT        0181  40  198#
STARET          02E5  396#  404  1007  1078  1253
SUBDH           056B  819#  879
TAB             0009  234#  369
TABO            02C1  371#  374
TABOUT          02BB  367#  383  387
TBUFF           0080  460#  1208
TESTVECTOR      054D  797#  1136
TRACK           08CE  619  1446#  1549
TRANV           08B8  531  633  1406#
TRUE            00FF  452#  541  1179
UBYTES          000D  656#  974
VRECORD         08C1  668  728  752  765  1078  1096  1420#
WORD            0002  456#  1349  1350  1356  1357  1358  1359  1360  1362  1363
                      1364  1365  1366  1368  1388  1392  1393  1394  1395  1396
                      1406  1425  1428  1444  1446  1447
```

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

COPYRIGHT © 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____